A Proposal for a Biometric Key Dependent Cryptosystem

GJCST Classification (FOR) D.4.6

K. Hassanain[1] ,M. Shaarawy, E. Hesham[2]

***Abstract*-With the increasing reliance on electronic information, which needs to be exchanged across the internet or stored on open networks, cryptography is becoming an increasingly important feature of computer security. A biometric key dependent cryptosystem is proposed, to ensure the security of the whole system by using fingerprint features as a key in a cryptosystem, like, key-dependent Advanced Encryption Standard (KAES). KAES is used to ensure that no trapdoor is present in cipher and to expand the key-space to slow down attacks.**

Keywords- AES, KAES, MD5, RNG, PRNG, SHA-1.

I. Introduction

Cryptography is becoming an increasingly important feature for information security, and there are many available cryptographic algorithms for securing information: Symmetric and Asymmetric. The strength of cryptosystem depends on many factors: key length, algorithm complexity and resistance to cryptanalysis techniques [1][2].There are mainly two problems when using traditional password or token as a key for any cryptosystem. First, the security of the key, and hence the cryptosystem, is now only as good as the password. Due to practical problems of remembering various passwords, some users tend to choose simple words, phrases, or easily remembered personal data, while others resort to write the password down on an accessible document to avoid data loss. The second problem is the lack of a direct connection between the password and the user, as a password is not tied to a user, a system running the cryptographic algorithm is unable to differentiate between the legitimate user and an attacker who fraudulently acquires the password of a legitimate user (Authentication) [1].An alternative to password protection, there are many approaches to bind a crypto-key with biometrics. The famous two approaches are a biometric-based key release and biometric-based key generation [2][3]. In biometric-based key release the key is hidden into a biometric template at the enrollment phase and is available to be released at authentication phase. While the other, the key is generated directly from the biometric data using one of secure hash functions [4].This paper introduces a biometric cryptosystem in which the key is generated from biometric data and produced key is used in key dependent encryption algorithm to ensure the security of the system and slow down its attacks. The paper is organized as follows: Section II presents the proposed biometric key dependent cryptosystem. Section III explains the evaluation criteria. Section IV discusses theexperimental results. Section V summaries and concludes the paper. References are given in Section VI.

About[1]- Faculty of computers and Information, Helwan University, Egypt. (email: mhmdshaarawy@yahoo.com, Hesham.eman@gmail.com
About[2]- Technical Research Department, Egypt(email: khass@idsc.net.eg

II. A Biometric Key Dependent Cryptosystem

The proposed scheme replaces the secret key in a cryptosystem with a key which generated directly from one of the human biometric data (e.g. fingerprint). In general, the proposed biometric-key cryptosystem could be subdivided into three phases: biometric phase, key generation phase and encryption phase. Figure 1 shows the overall structure of the proposed system.

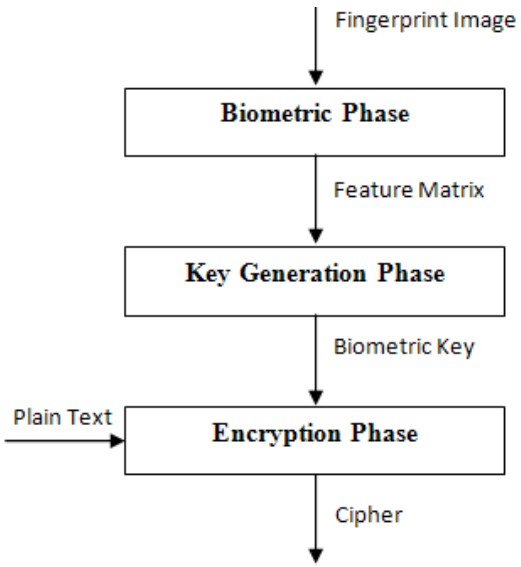

Fig.1. The proposed biometric key dependent cryptosystem

In the proposed system, the input to the biometric phase is fingerprint image which acquired from the system user's finger using fingerprint reader. Through this phase some unique characteristics of the fingerprint image are extracted to form a biometric feature matrix.The produced matrix is used as an input to the next phase to generate a 128-bit key using one of cryptographic hash functions such as Secure Hash Algorithm (SHA-1) or Message-Digest algorithm 5 (MD5). The plain-text is then encrypted using the generated key by one of cryptographic encryption algorithm such as AES or KAES.Each phase of the proposed system is described in more details in the subsections.

1) Biometric Phase

A fingerprint represents a pattern of ridges and valleys on the person finger's tip. And also can be defined by the uniqueness of the local ridge characteristics and their relationships. Figure 2 shows these characteristics by marking minutiae points for the finger image. Minutiae points are these local ridge characteristics that occur either at a ridge ending or a ridge bifurcation [6].In low quality fingerprint images which contain noise and contrast deficiency usually results in pixel configurations similar to minutiae points. So, the automatic minutiae detection process becomes a difficult task [6].

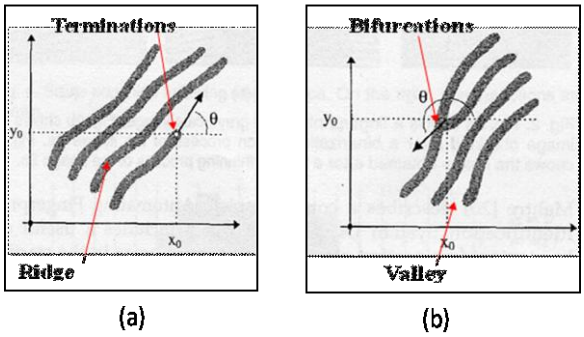

Fig.2. Fingerprint: (a) Termination minutia. (b) Bifurcation minutia.

The following steps are run in the biometric phase to form a biometric feature matrix using the input fingerprint image [6]:

1. Pre-Processing.
2. Minutiae extraction.
3. Post-Processing.

The first step called pre-processing and runs different tasks to enhance the input fingerprint image. The next step deals with the extraction of minutiae. In the third step false minutiae are deleted from the set of minutiae which obtained early. *Pre-Processing* step contains three different stages namely, image enhancement, image binarization and image segmentation. Each stag runs one or more different image processing methods as shown in figure 3. The input for this step is the fingerprint image and the output is an enhanced fingerprint image that is a suitable input for the following minutiae extraction step [6]. At *Minutiae extraction* step, the skeleton of the image is formed and the minutiae points are then extracted by the following method [6]:1. The binary image is thinned as a result of which a ridge is only one pixel wide.2. The minutiae points are thus those which have a pixel value of one (ridge ending) as their neighbor or more than two ones (ridge bifurcations) in their neighborhood

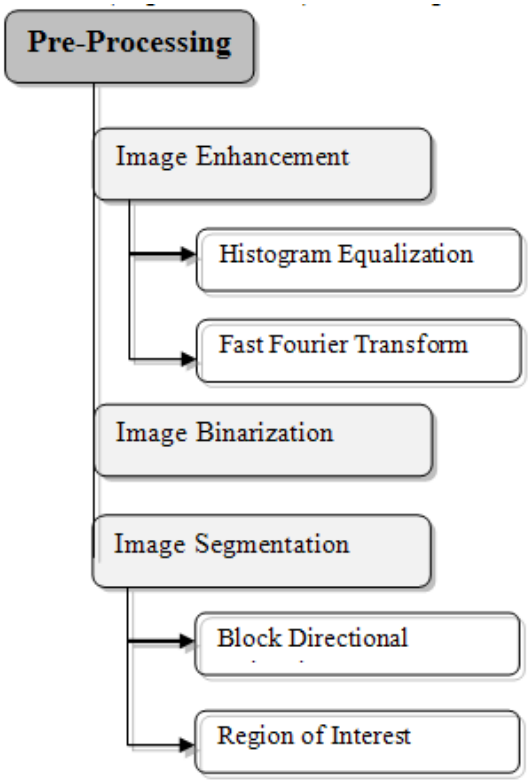

Fig.3. Pre-Processing Step

The output of this step is an Nx3 biometric feature matrix which contains x positions, y positions and orientations for N minutiae as:

X position	Y position	Orientation
112.0000	77.0000	3.1416
208.0000	34.0000	-1.3191
50.0000	54.0000	0.1326
177.0000	73.0000	2.4585
39.0000	55.0000	0.0286
157.0000	239.0000	-2.3816

Post-Processing step removes the false minutiae from the biometric feature matrix. These false minutiae may occur due to the presence of ridge breaks in the given image itself which could not be improved even after enhancement. Figure 4 shows different situations for false minutiae points.

The biometric phase outputs an Nx3 matrix which usually holds the true N minutiae points' information.

The biometric phase outputs an Nx3 matrix which usually holds the true N minutiae points' information.

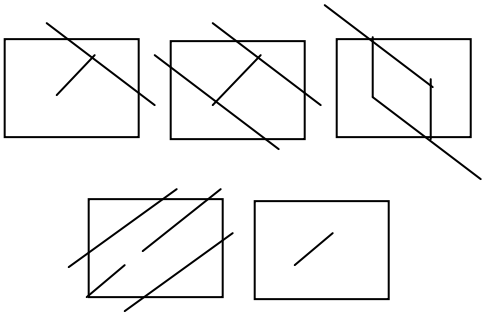

Fig.4. False Minutiae Points

The biometric phase outputs an Nx3 matrix which usually holds the true N minutiae points' information.

Key Generation Phase

In the proposed, MD5 is used to generate 128-bit encryption key from the generated biometric feature matrix .MD5 algorithm consist of 5 steps namely, Append Padding Bits, Append Length, Initialize MD Buffer, Process Message in 16-Word Blocks and Finalize the Output [7].The output generated key from MD5 is suitable for many encryption algorithms like AES and KASE.

Encryption Phase

KAES is a symmetric encryption technique that changes AES to be key dependent techniques. KAES is block cipher in which the block length and the key length are specified according to AES specification: 128, 192, or 256 bits and block length of 128 bits. In the proposed system, a key length of 128 bits is used. KAES involves the key in most of algorithm steps which increase the security of it rather than in AES. The main differences between AES and KAES can be summarized as in Table 1[5].Through the encryption phase, KAES has been applied to encrypt the plain-text using the generated key.

	AES	KAES
Round Function	last round is different	The same transformations in all rounds.
S-BOX	Fixed	S-Box is key dependent
Key Expansion	Fixed S-Box	Generate d S-Box
Round Transformation	Independent on the key	Dependent on the key
Shift Offset	Use Fixed from 0 to 3	Reliant on the key

Table 1. Differences between KAES and AES

III. Evaluation Criteria

For evaluating randomness of the proposed system, various tests were applied. To facilitate interpretation of the experimental results, a brief description is given, to make the analysis of these tests output understandable.

1) Entropy

Entropy describes the number of bits per byte. It is known as information density for a file. Extremely entropy output indicates that information is essentially random. Hence optimal compression is unlikely to reduce its size.

2) Optimal "Best" Compression

Reflects compressibility and is computed based on entropy encoding. For Example, if a file has 4.9 as Entropy, the optimal compression (OC) of the file would reduce its size by 38% as follow:

OC = 100 – (Entropy / 8)*100

OC = 100 – 62 = 38 %

3) Chi-square Distribution

The randomness of data can be tested using the chi-square test. The chi-square distribution is as an absolute number and a percentage which indicates how frequently a truly random sequence would exceed the value calculated.The percentage is interpreted as the degree to which the sequence tested is suspected of being non-random as:

- If the percentage is greater than 99% or less than 1%, the sequence is almost certainly Not Random.
- If the percentage is between 99% and 95% or between 1% and 5%, the sequence is Suspect.
- Percentages between 90% and 95% and 5% and 10% indicate the sequence is Almost Suspect.
- Otherwise the sequence is random.

4) Arithmetic Mean Value

Arithmetic mean value for a sequence of data is simply the result of summing the all the bytes and dividing by the sequences length. If the data are close to random, this should be about 127.5. If the mean departs from this value, the values are consistently high or low.

5) Monte Carlo value for PI

Each successive sequence of six bytes is used as 24-bit x and y coordinates within a square. If the distance of a randomly generated point is less than the radius of a circle inscribed within the square, the six byte sequence is considered a "hit". The percentage of hits can be used to calculate the value of PI. If the computed value approaches the correct value of PI, the sequence is close to random. Monte Carlo value for Pi is 3.143580574 (error 0.06 percent).

6) Serial Correlation Coefficient

The quantity measures the extent to which each byte in a sequence depends upon the previous byte. If the value (which can be positive or negative) close to zero, the

sequence is random (totally uncorrelated). Otherwise serial correlation coefficient will be greater than or equal 0.5.

IV. Experimental Results

To simulate the proposed biometric key dependent cryptosystem, a MATLAB script was implemented for biometric phase and for AES and KAES also a java program was implemented for key generation phases. The key's length (128 bit) was fixed for both AES and KAES algorithms. Fingerprint image (fingerprint3.tif) which captured by Cross Match Verifier 300 scanner at 500 dpi, is used to test our system. Figure 5 shows the inputs and outputs of the biometric phase program. The minutiae points are saved as a biometric feature matrix.

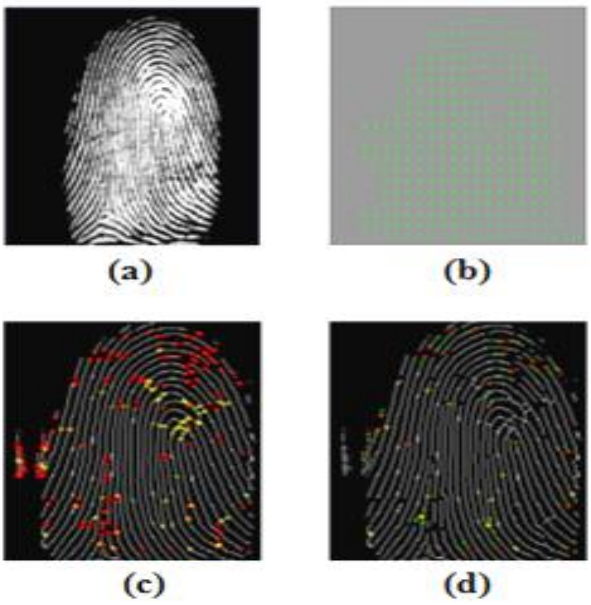

Fig.5. Biometric Phase: (a) Input Image (b) Preprocessing output (c) Extraction output (d) Post processing output

The generated 128 key from the biometric features matrix through the key generation phase is:
0X9e 0Xf8 0X68 0X62
0X53 0Xdf 0X3c 0Xc5
0Xf2 0X66 00Xef 0Xde
0Xb6 0Xc7 0Xbe 0X3a

For the testing another plain key is used:
0Xb1 0Xc2 0Xf3 0X84
0X75 0Xa6 0Xd7 0X08
0X19 0X13 0X11 0X42
0X53 0X20 0X15 0X16

Table 2 lists six files with their format and sizes. These files are used in encryption and decryption steps. Using ENT [8] results is obtained by carrying out the evaluation criteria discussed in section 3.

	Name	**Size** (Bytes)	**Name**
Text	secret_t1.txt	2815	T1
	secret_t2.txt	4007	T2
Audio	secret_w1.wav	14077	W1
	secret_w2.wav	35191	W2
Image	secret_im1.tif	95162	Im1
	secret_im2.tif	1001648	Im2

Table 2. Files Names

Table 3 illustrates the occupation of bits within a byte, it could be noticed that KAES and biometric KAES utilizes almost every bit in a byte, introducing extremely dense files.Figure 6 depicts the optimum "best compression" that can be achieved. The results are the same for some of the experimented files.

The Chi-Square distribution for the experimental file is shown in figure 4. Also Chi-Square distribution gives a good indication of the proposed system randomnTable 4 shows the computed arithmetic mean for both AES and KAES are close to the arithmetic mean value = 127.5.Table 5 Shows the computed Monte-Carlo values of piwhich are close to the value of PI= 3.1416.Figure 8 indicates that relation between successive bytes is very small as Serial Correlation Coefficients

.

	T1	T2	W1	W2	Im1	Im2
Original	4.711491	5.166631	6.215379	5.963367	5.910927	7.562718
KAES_PlainKey	7.911388	7.952993	7.98213	7.986803	7.996512	7.999835
KAES_Fingerprint key	7.930583	7.947021	7.983541	7.985064	7.996409	7.99978
AES_PlainKey	7.926212	7.957024	7.984306	7.983461	7.997254	7.999767
AES_Fingerprint key	7.918981	7.955877	7.985193	7.985132	7.996956	7.999767

Table 3. Entropy Values

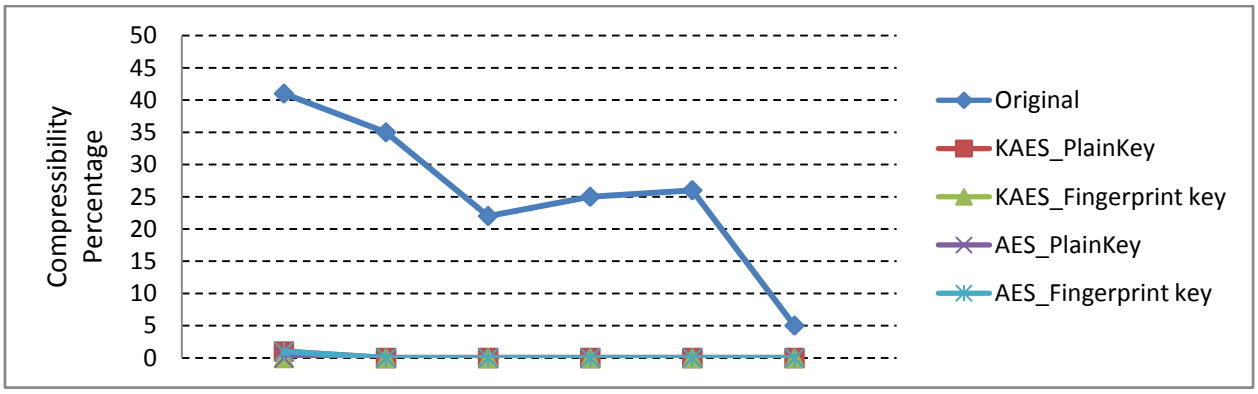

Fig.6. The Optimum Compression

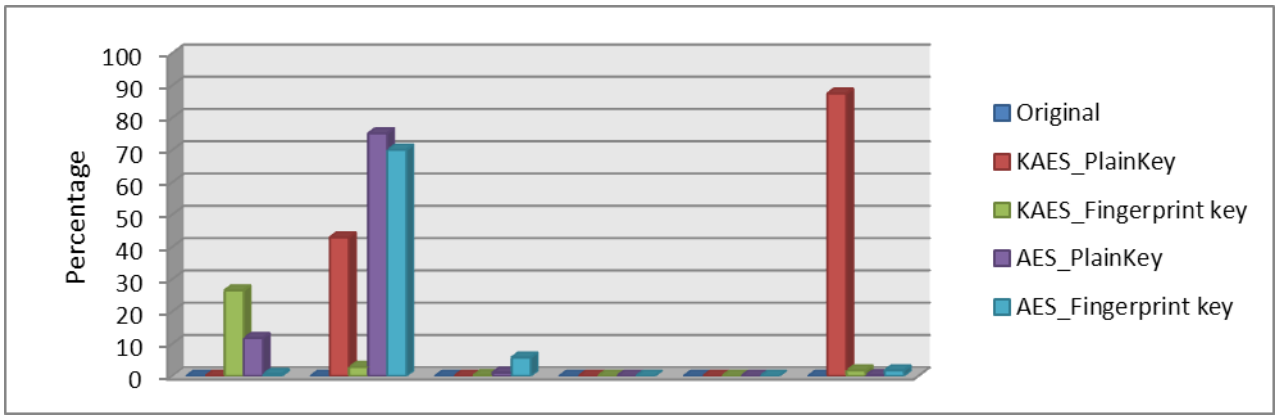

Fig.7. The Chi-Square distribution

	T1	T2	W1	W2	Im1	Im2
Original	85.5844	133.5031	127.0035	121.4201	65.9731	93.1652
KAES_PlainKey	127.4348	124.9963	127.1517	127.1777	127.0659	127.5825
KAES_Fingerprint key	127.0274	128.4168	127.5046	128.2845	127.469	127.5014
AES_PlainKey	130.323	127.2298	127.7911	128.4338	127.58	127.6038
AES_Fingerprint key	128.6254	128.1158	127.7079	127.3429	127.4235	127.6049

Table 4. The Computed Arithmetic Mean

	T1	T2	W1	W2	Im1	Im2
Original	4	2.824587706	3.884057971	3.957033248	3.384867591	3.865725017
KAES_PlainKey	2.950959488	3.166416792	3.15942029	3.156351236	3.15889029	3.136844754
KAES_Fingerprint key	3.138592751	3.274362819	3.15771526	3.125660699	3.161916772	3.14951989
AES_PlainKey	3.068376068	3.076461769	3.109974425	3.144075021	3.157124842	3.136293661
AES_Fingerprint key	3.025641026	3.148425787	3.15771526	3.118158568	3.145775536	3.136293661

Table 5. The Monte-Carlo Values

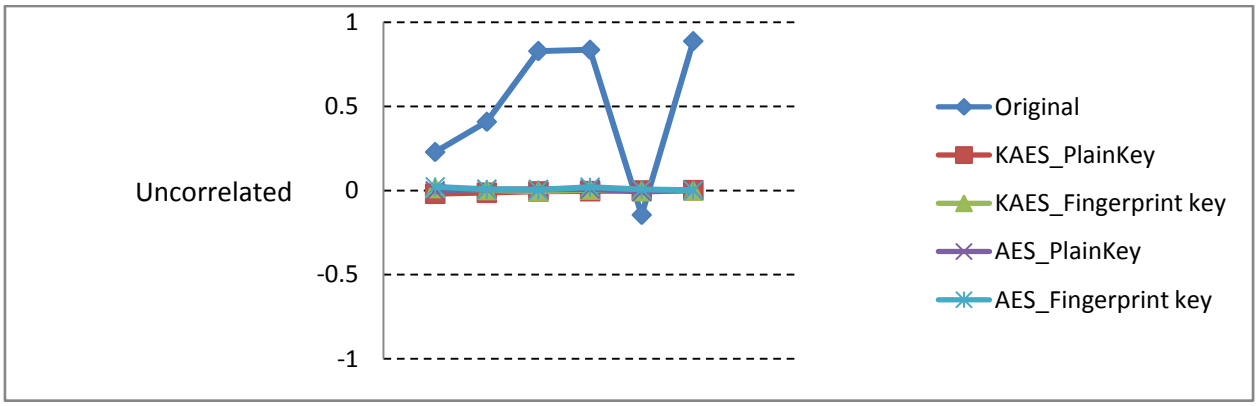

Fig.8. The Serial Correlation Coefficient

V. Conclusion

This paper presents a biometric key dependent cryptosystem by replacing the plain key with fingerprint feature data. KAES is improving the security of the proposed system by employing the key to be the main parameter of the encryption algorithm. Experiments analysis for biometric and key generation phases will be reported in the near future to insure the reliability and the security of the proposed system.

VI. References

1) International Computer Security Association., & Nichols, R. K. (1999). ICSA guide to cryptography (chapter 22). New York: McGraw Hill.
2) Stoianov, A., Information and Privacy Commissioner /Ontario., & Cavoukian, A. (2007). Biometric encryption: A positive-sum technology that achieves strong authentication, security and privacy. Toronto, Ont: Information and Privacy Commissioner, Ontario.
3) Kresimir & Mislav(2004, June). A survey of biometric recognition methods. presented at 46th International Symposium Electronics in Marine, ELMAR-2004
4) Li, W., Zhan, C., & Zheng, G. (January 01, 2006). Cryptographic Key Generation from Biometric Data Using Lattice Mapping. Proceedings, 513-516.
5) Uludag, U. (2006). Secure biometric systems.
6) Nimitha Chama(2003). Fingerprint Image Enhancement and Minutiae Extraction. University of Clemson.
7) Network Working Group, R. Rivest & RSA Data Security. Retrieved July 25, 2010 from Internet FAQ Archives Web site: http://www.faqs.org/rfcs/rfc1321.html
8) ENT. Retrieved July 20, 2010 from A Pseudorandom Number Sequence Test Program Web site: http://www.fourmilab.ch/random